Security-Focused Training Model of Reinforcement Learning in Autonomous Vehicles

Takahito Yoshizawa*, Alireza Aghabagherloo*, Árpád Huszák†, Csongor Ujvárosi†, Dave Singelée*, Bart Preneel*

*COSIC, KU Leuven,
Kasteelpark Arenberg 10 Bus 2452, B-3001 Leuven, BELGIUM
{takahito.yoshizawa, alireza.aghabagherloo, dave.singelee, bart.preneel}@esat.kuleuven.be

†Budapest University of Technology and Economics (BME),
Faculty of Electrical Engineering and Informatics, Department of Networked Systems and Services
Budapest, Műegyetem rkp. 3, 1111 Hungary
huszak@hit.bme.hu, ujvarosi.csongor@edu.bme.hu

***Abstract*—Reinforcement Learning (RL) in autonomous vehicles (AVs) is expected to enhance the safe maneuvering of AVs to improve road safety. However, existing literature on AVs focuses on the impacts of image perturbations as adversarial examples (AEs) during the testing phase. Limited attention has been given to more intrusive types of AEs, such as vehicles with adversarial intent to induce accidents on the road proactively. Without addressing this type of AEs, the learned policy remains vulnerable to different types of AEs, making the RL unusable in AV networks, given the nature of cyber-physical systems (CPS), for which negative consequences include accidents, property losses, and injuries. We focus on the training phase to address this gap and fortify the learned policy using our expanded AE definitions. This paper presents our approach to realizing this training model to build a more robust policy against adversaries.**

***Index Terms*—Reinforcement Learning, Deep Reinforcement Learning, Machine Learning, Security, Autonomous Vehicle, Cyber-Physical System (CPS)**

I. Introduction

Machine Learning (ML), specifically Reinforcement Learning (RL), is considered a key technology to make autonomous vehicles (AVs) a reality [1]. Research on RL has advanced in the last few decades, resulting in much literature published on this subject. However, the focus on security aspects of RL on AVs has gained a minor part of research in this context. Existing literature in this area mainly focused on its vulnerabilities against adversarial examples (AEs), such as perturbations of visual images and their impact on AVs. Examples include applying small changes to a stop sign to cause AVs to misinterpret it [2]. Others include incorrect lane detection with the presence of adversaries [3] or without them [5].

This work was supported in part by CyberSecurity Research Flanders with reference number VR20192203, and the CELSA project "Machine Learning-Based Cooperative Vehicle and Traffic Control Using Secure ML and V2X Communications (ML-CCAM)." This project is also partially supported by the AIDE project funded by the Belgian FOD BOSA, DG Digital Transformation with reference nr. 06.40.32.33.00.10.

From our study of existing literature, we observe that research work that directly addresses the vulnerability of RL policies deserves attention – new approaches that improve the learned policy by making it more robust against AEs in practical applications such as AVs. In this context, we propose a new RL training approach. Our underlying hypothesis is that, by taking the presence of AEs and their behavior into consideration during the training phase, the resulting policy will become more robust against AEs when it is deployed on the road. This paper presents our training model and simulation plan to validate this hypothesis.

The rest of this paper is organized as follows. We first cover background and related work in Sec. II; we then discuss our approach to the stated problem and the resulting methodology to conduct our work in Sec. III, followed by our learning model in Sec. IV. Then, we conclude our discussion in Sec. V with next steps to follow through with our plan.

II. Background and Related Work

A. Reinforcement Learning

Reinforcement learning sits at the nexus of machine learning and optimal control theory, offering a powerful framework for agents to learn optimal decision-making policies in dynamic environments. The RL paradigm centers around an agent interacting with its environment. The agent perceives the environment through observations and takes actions that influence the environment's state. This interaction follows the Markov Decision Process (MDP) framework, where the current state holds all relevant information about past interactions. An agent's policy maps observations to actions, defining its decision-making strategy. Policy is defined as the probability distribution over actions for every possible state.

$$\pi\left(a|s\right) = P\left(\mathcal{A} = a|\mathcal{S} = s\right) \tag{1}$$

RL methods specify how the agent changes its policy due to its experience. The optimal policy, denoted as $\pi^* : \mathcal{S} \rightarrow \Delta\mathcal{A}$,

is the policy that leads to the highest expected total reward. The notation $\Delta\mathcal{A}$ represents probability distributions over the set of actions ($\mathcal{A}$). Building upon the core principles of RL, Deep Reinforcement Learning (DRL) leverages deep neural networks for complex scenarios where traditional RL struggles. Deep Q-Network (DQN) and Proximal Policy Optimization (PPO) are the most used DRL algorithms. While the value-based DQN estimates the Q-value function, representing the expected future reward for taking an action in a given state, PPO is an actor-critic method combining elements of both value-based and policy-based approaches. Although PPO offers advantages over DQN in terms of stability, sample efficiency, and handling continuous control tasks [4], PPO is not guaranteed to give better results in all scenarios.

B. Related Work

As discussed in the previous Sec. I, much of the existing research work on the security aspect of RL in the AV context discusses visual perturbation on road signs and markings. It describes their consequence on the AVs' perceptions, such as misinterpreted stop signs or lane markings [2], [3], [5]. Several surveys focus on using DRL in the AV context [6]. However, coverage of its security aspect is either minimal or non-existent. From the collision perspective, Behzadan and Munir demonstrated, in a simulated setting, that an AE vehicle intentionally caused collisions in several scenarios [7]. From a robotics perspective, Mohanan and Salgoankar categorized and itemized many approaches for motion planning and collision avoidance [8]. Several literature propose schemes to improve the robustness of DRL by incorporating AEs during the training process [9]–[11]. However, their concept is demonstrated using OpenAI Gym MuJoCo; its applicability to practical usages such as AVs is not covered; thus, it is an open question. Based on our observation, our security-focused training model is a novel approach to making RL secure and robust.

III. Methodology

A. Context

We consider an intersection without a traffic light, as shown in Fig. 1. In this scenario, we consider a safe passage of an AV of interest (an *ego vehicle*) through this intersection when other vehicles are present near the intersection. We take several incremental steps with increasing complexities for the *ego vehicle* to determine its maneuver as it approaches the intersection. As the first step, we take V2X messages from all vehicles as the input data.[1] As all vehicles within the communication range receive other vehicles' messages, each vehicle establishes and maintains the *situational awareness* of the traffic around it. As the next step, we add other sensor data, such as camera, radar, lidar, etc., for more complex decision-making for maneuvering.

[1]In Europe and the US, Cooperative Awareness Message (CAM) is defined in ETSI TS 302 637-2 [12] and Basic Safety Message defined in SAE J2735 [13] are specified, respectively. Each vehicle broadcasts this message to announce its position, speed, direction, acceleration, etc.

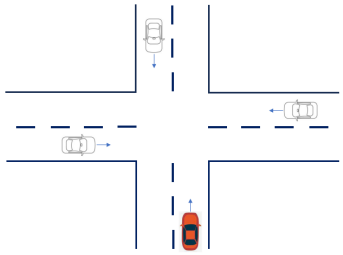

Fig. 1. Maneuvering at an Intersection without Traffic Light

B. Adversarial Model

We define our adversarial model so that the adversaries aim to cause traffic accidents with varying characteristics and capabilities. To this end, we define two dimensions of adversary types: *active* vs. *passive*, and *evident* vs. *non-evident* adversaries.[2]

***Active* vs. *Passive* Adversaries:**

1) *Active* adversary: This adversary proactively triggers actions to cause accidents. It considers the *ego vehicle*'s presence upon determining its movement, such as intentionally making sudden changes to its movement or adjusting its speed to block the *ego vehicle* from passing the intersection safely.
2) *Passive* adversary: This adversary uses a fixed and predetermined maneuvering rule and does not consider *ego vehicle*'s presence. Its goal is still to cause accidents. However, its actions are more subtle than *active* adversary; it does not exhibit observable *abnormal* behavior in its maneuver. Therefore, it does not expose adversarial intent to the *ego vehicle*, maintaining the deniability of its adversarial intent by staying *inactive*.

***Evident* vs. *Non-evident* Adversaries:**

1) *Evident* adversary: This adversary openly exhibits its adversarial intent in its maneuver to cause accidents. Examples include *dangerous* behavior, such as violating the speed limit or providing incorrect information in its maneuver, such as indicating a left turn while turning right. In this sense, other vehicles can potentially observe adversarial intent from this type of adversary.
2) *Non-evident* adversary: This adversary behaves within the boundary of an honest vehicle's behavior. It follows the traffic rules, such as the speed limit, and sends benign information. In this sense, this adversary appears to be an honest vehicle. However, its essential characteristic is sending incorrect information while appearing correct. This way, its behavior is more difficult for other vehicles to detect and prove that it is an adversary.

The resulting two combinations are (1) *active-evident* adversaries and (2) *passive-non-evident* adversaries. Given that we have only V2X communication as input data in the initial step,

[2]The difference between *active* and *evident* adversaries are subtle. The former is concerned with intrusiveness to the environment, and the latter is concerned with the visibility of the behavior to other vehicles.

it is not possible to combine and correlate with other inputs, such as camera images. Therefore, *active-evident* adversaries have a higher chance of causing accidents as they intentionally mislead the *ego vehicle*. As a result, it is more challenging to robustify the RL policy against this adversary type. On the other hand, *passive-non-evident* adversaries are more discreet and subtle in their behavior, i.e. *deniability* of their actions. Therefore, the robustified policy would be more effective against them. This is our hypothesis.

C. Robust Policy Learning Model

We define our approach to the simulation work based on the adversary model definition in the preceding section. We follow the steps to make the solution with an incremental level of sophistication and robustness against different adversaries: (step 1) introduce more *normal* AEs based on *active-evident* adversaries. Here, *normal* means adversaries' behavior is more obvious *bad behavior* to cause accidents, (step 2) introduce *passive-non-evident* adversaries, which are easier for the *ego vehicle* to avoid accidents, (step 3) design a robust model against step 2 above, (step 4) design a robust model against step 1 above. With this approach, we tackle the problem incrementally in the order of more accessible to more complex levels of robustness to design security solutions against various adversary types. This strategy is illustrated in Fig. 2.

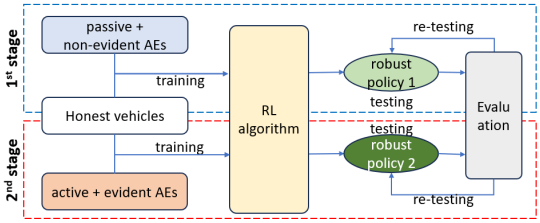

Fig. 2. Robust Policy Learning Model

D. RL Formulation and Simulation Set Up

To implement the intersection environment without traffic lights, we use the SUMO urban mobility simulator [14] and its traffic control interface (TraCI) [15] to retrieve values from the simulation and control vehicles' behavior. We use the multi-agent RL (MARL) concept that allows cooperative agents to control multiple *ego vehicles* in the simulated environment. The aim is to learn how to adaptively adjust the speed to avoid collisions with both normal and adversary vehicles.

To implement the reinforcement learning environment, we use the Farama Gymnasium framework (formerly OpenAI Gym) [16] that provides a standardized environment structure compatible with different RL Python libraries. Besides implementing the intersection topology and traffic demands, the RL environment requires the definition of observation space, actions space, and reward function. The following concept is used in the proposed autonomous intersection model:

Obesrvation space. To learn good decisions in different situations, the *ego vehicles* monitor their environment to collect information about other vehicles within the intersection area. We assume V2X communication that assures that each vehicle broadcasts sensor data that is received by all vehicles within the intersection zone. As a first step, we rely only on $speed$ and x, y coordinates of vehicles due to the limitation of the SUMO simulator. RL algorithms handle only fixed-length observation spaces, so we maximize the number of agent-controlled *ego vehicles* ($|N^{AC}|$) and non-agent-controlled vehicles ($|N^{nAC}|$) in the intersection by using a fixed-sized state vector (S_i) for each agent. A zero-filling approach is used if fewer vehicles are in the control zone. The first values in the state vector are related to i^{th} *ego vehicle* and followed by collected data from other *ego vehicles* ($j \in N^{AC}$ and $j \neq i$) and non-agent-controlled vehicles ($k \in N^{nAC}$). Non-agent-controlled vehicles can behave as regular vehicles or different types of adversary vehicles. The formulation of the agent state is as follows:

$$s = \{v, x, y\} \tag{2}$$

$$S_i = \left\{ s_i, (s_j)_{\, j \in N^{AC}, j \neq i}, (s_k)_{\, k \in N^{nAC}} \right\} \tag{3}$$

An advanced version of the observation space is where not just the state of the current timestep is included but also a few previous states. Using multiple timeframe information can help the agent policy to estimate vehicle motions and determine more effective actions. Moreover, in the next phase of our research work, we will use a more sophisticated simulator (e.g., CARLA [17], SMARTS [18]) to include other sensor data in the observation space, such as camera, lidar, and radar.

Action space. The state vectors for each *ego vehicle* serve as the input of the RL algorithm, while the outputs are the agent actions. The possible actions must be pre-defined as a set of actions $a \in \mathcal{A}$. In the current phase of the work, we define three possible actions $\mathcal{A} = \{v^+, 0, v^-\}$: increase, keep, and reduce the speed. By replacing SUMO with another more complex simulator, the action space can also be extended by steering actions.

Reward. The reward function is the most critical factor from the training efficiency point of view. The agents aim to perform a series of actions that lead to maximum cumulated reward. In the autonomous intersection scenario, the goal is to avoid collision with other agent-controlled and regular vehicles as well as adversary vehicles that try to hit *ego vehicles*. Thus, the reward value received by an agent in each timestep is defined as follows:

$$R = \begin{cases} -100 & \text{penalty for collision} \\ -1 & \text{small penalty in each step to encourage fast crossing} \\ 100 & \text{leaving the intersection zone} \end{cases} \tag{4}$$

IV. Learning Model with AEs

This model expresses the idea of our approach to taking security into RL learning.

A. Training Phase

We aim to investigate how the presence of adversary vehicles influences the reliability of the trained model and its performance in the testing phase. As defined in the observation space of RL agents, non-agent-controlled vehicles can be normal-behaving regular vehicles (honest) or different types of adversary vehicles ($N^{nAC} = M \cup O$), where M represents AE vehicles and O denotes the set of honest, non-agent-controlled vehicles. Thus, the total vehicle population set can be formulated as $N = N^{AC} \cup M \cup O$. Hence, $\sigma = \frac{|M|}{|N|}$ represents the % of AEs in the vehicle population.

Agent's state relies on CAM messages broadcasted by each vehicle as formalized in (2) and (3). However, some of the non-agent-controlled vehicles might be adversaries sharing invalid information by manipulating the sensor data.

$$x \leftarrow x + \delta : x \in s, \quad \text{where } \delta = \begin{cases} 0 & \text{if } veh \notin M, \\ \neq 0 & \text{if } veh \in M, \end{cases} \tag{5}$$

where x and δ represent a CAM message and its perturbation.

B. Testing Phase Using the Learned Policy

The testing phase result using the learned policy is:

$$Res = DT(\pi), \quad \text{where } DT(\pi) = \begin{cases} 0, & \text{if failure,} \\ a, & \text{if near miss,} \\ 1, & \text{if success.} \end{cases} \tag{6}$$

where $DT(\pi)$ and Res are drive test using a policy π and its result, respectively. Success means the *ego vehicle* safely passed the intersection without collision. Failure means the *ego vehicle* had a collision. The value a, where $(0 < a < 1)$ represents that the vehicle passed through the intersection without collision, but there was/were vehicle(s) nearby (near miss situation), thus was close to having an accident. The actual value of a can be determined based on a function of how close and how many vehicles were involved.

C. Varying % of AEs and the Policy's Effectiveness

We run multiple iterations of both training and testing phases with varying % of AEs to see the resulting effectiveness of the learned policies. Here, Θ represents the training process based on a specific % of AEs ($0 \leq \sigma_i \leq 1$) in the vehicle population. The resulting learned policy is represented as π_i.

$$\pi_i \leftarrow \Theta(\sigma_i) \tag{7}$$

We evaluate the effectiveness of the training for each π_i by testing its policy in the testing phase.

$$\bar{E} = \{E(\pi_0), E(\pi_\cdot), E(\pi_2), E(\pi_3)...\}, \tag{8}$$

where $E(\pi_i)$ represents the effectiveness of the learned policy for σ_i from the perspctive of safe passage through the intersection. At the end of our simulation and evaluation, we expect to complete a two-dimensional table with varying % of AEs during the training and testing phases.

V. Next Steps

Our simulation work based on the methodology discussed in Sec. III is a work in progress. Completing the trial runs and obtaining and analyzing results is our first step in future work. Based on this analysis, we evaluate training effectiveness and further testing phases. Furthermore, we will explore further improvement of our training methodology to improve its effectiveness against AEs.

References

[1] J. R. N. Forbes, "Reinforcement learning for autonomous vehicles", PhD. thesis, University of California, Berkeley, 2002

[2] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, D. Song, "Robust physical-world attacks on deep learning visual classification", Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1625–1634, 2018

[3] P. Jing, Q. Tang, Y. Du, L. Xue, X. Luo, T. Wang, S. Nie, S. Wu, "Too good to be safe: Tricking lane detection in autonomous driving with crafted perturbations", 30th USENIX Security Symposium (USENIX Security 21), pp. 3237–3254, 2021

[4] S. Kabanov, G. Mitiai, H. Wu, and O. Petrosian, "Comparison of Reinforcement Learning Based Control Algorithms for One Autonomous Driving Problem," in International Conference on Mathematical Optimization Theory and Operations Research, 2022, pp. 338—349.

[5] T. Sato, J. Shen, N. Wang, Y. Jia, X. Lin, Q. A. Chen, "Dirty road can attack: Security of deep learning based automated lane centering under {Physical-World} attack", 30th USENIX Security Symposium (USENIX Security 21), pp. 3309–3326, 2021.

[6] S. Grigorescu, B. Trasnea, T. Cocias, G. Macesanu, "A survey of deep learning techniques for autonomous driving", Journal of Field Robotics, Wiley Online Library, vol. 37, no. 3, pp. 362–386, 2020

[7] V. Behzadan, A. Munir, "Adversarial reinforcement learning framework for benchmarking collision avoidance mechanisms in autonomous vehicles", IEEE Intelligent Transportation Systems Magazine, vol. 13, no. 2, pp. 236–241, 2019

[8] M. G. Mohanan, A. Salgoankar, "A survey of robotic motion planning in dynamic environments", Robotics and Autonomous Systems, Elsevier, vol. 100, pp. 171–185, 2018

[9] A. Pattanaik, Z. Tang, S. Liu, G. Bommannan, G. Chowdhary, "Robust deep reinforcement learning with adversarial attacks", arXiv preprint arXiv:1712.03632, 2017

[10] L. Pinto, J. Davidson, R. Sukthankar, A. Gupta, "Robust adversarial reinforcement learning", International Conference on Machine Learning, PMLR, pp. 2817–2826, 2017

[11] P. Zhai, J. Luo, Z. Dong, L. Zhang, S. Wang, D. Yang, "Robust adversarial reinforcement learning with dissipation in equation constraint", Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, no. 5, pp. 5431–5439, 2022

[12] European Telecommunications Standards Institute (ETSI), "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service", Ver.1.4.1, Apr. 2019

[13] Society of Automotive Engineers (SAE), "J2735 Surface Vehicle Standard, V2X Communications Message Set Dictionary", Nov. 2022

[14] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, E. Wießner, "Microscopic Traffic Simulation using SUMO", The 21st IEEE International Conference on Intelligent Transportation Systems, IEEE Intelligent Transportation Systems Conference (ITSC), 2018

[15] SUMO documentation, "Traffic Control Interface (TraCI)", [ONLINE] https://sumo.dlr.de/docs/TraCI.html

[16] M. Towers, J.K. Terry, A. Kwiatkowski, J.U. Balis, G. Cola, T. Deleu, M. Goulão, A. Kallinteris, M., et al., "Gymnasium", Zenodo, 2023, [ONLINE] https://gymnasium.farama.org/

[17] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, V. Koltun, "CARLA: An Open Urban Driving Simulator," in Proceedings of the 1st Annual Conference on Robot Learning, PMLR, pp. 1–16, 2017

[18] M. Zhou, J. Luo, J. Villella, Y. Yang, D. Rusu, J. Miao, W. Zhang, M. Alban, I. Fadakar, Z. Chen, "SMARTS: Scalable Multi-Agent Reinforcement Learning Training School for Autonomous Driving," arXiv preprint arXiv:2010.09776, 2020.